Audio-visual vibraphone transcription in real time

Tiago F. Tavares [#1], Gabrielle Odowichuck [*2], Sonmaz Zehtabi [#3], George Tzanetakis [#4]

[#] *Department of Computer Science, University of Victoria*
Victoria, Canada
[1] tiagoft@uvic.ca, [3] szhetabi@uvic.ca, [4] gtzan@cs.uvic.ca

[*] *Department of Electrical and Computer Engineering, University of Victoria*
Victoria, Canada
[2] godowichuk@gmail.com

***Abstract*—Music transcription refers to the process of detecting musical events (typically consisting of notes, starting times and durations) from an audio signal. Most existing work in automatic music transcription has focused on offline processing. In this work we describe our efforts in building a system for real time music transcription for the vibraphone. We describe experiments with three audio-based methods for music transcription that are representative of the state of the art. One method is based on multiple pitch estimation and the other two methods are based on factorization of the audio spectrogram. In addition we show how information from a video camera can be used to impose constraints on the symbol search space based on the gestures of the performer. Experimental results with various system configurations show that this multi-modal approach leads to a significant reduction of false positives and increases the overall accuracy. This improvement is observed for all three audio methods, and indicates that visual information is complimentary to the audio information in this context.**

***Index Terms*—Audiovisual, Music, Transcription**

I. Introduction

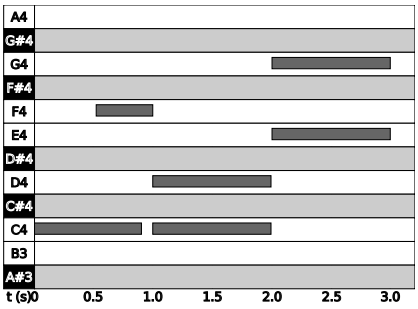
There is an increasing trend of interfacing musical instruments with computers. The most common approach is to create specialized digital instruments that send digital signals about what is played using protocols such as the Musical Instrument Digital Interface (MIDI) or Open Sound Control (OSC) protocols. However, these instruments frequently lack the haptic response and expressive capabilities of acoustic instruments. A more recent approach has been to retrofit acoustic instruments with digital sensors to capture what is being played. These so called "hyperinstruments" [1] combine the haptic feel and expressive capabilities of acoustic instruments while providing digital control information. However, they tend to be expensive, hard to replicate, and require invasive modifications to the instrument which many musicians dislike. Indirect acquisition [2] refers to the process of acquiring the control information provided by invasive sensors by analyzing the audio signal acquired by a microphone. When successful such systems can provide similar capabilities to hyperinstruments without requiring invasive modification. They are also easy to replicate, and have significantly lower cost.

MMSP'10, October 4-6, 2010, Saint-Malo, France.
???-?-????-????-?/10/$??.??

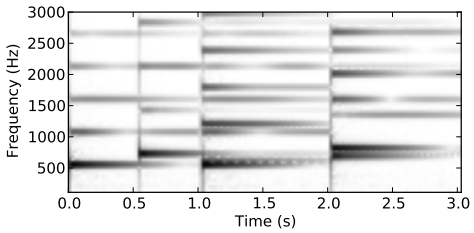
The focus of this paper is the indirect acquisition of control information for the vibraphone which is a member of the pitched percussion family. The major components of a vibraphone are aluminum bars, tube resonators placed under the bars, and a damper mechanism that controls whether notes are sustained after they are struck or not. There are some digital controllers with similar layout to a vibraphone that simply trigger digital samples and do not produce acoustic sound. It is also possible to augment acoustic vibraphones with digital sensors to obtain information about what bar is being played and when. However, this is costly and hard to accomplish without altering the acoustic response of the instrument. In this paper we describe how information about what bar is being played and when can be extracted by computer analysis of audio-visual information. The audio signal acquired by a single microphone, that is not attached to the instrument, is used as input to automatic transcription algorithms. Such algorithms rely on digital signal processing techniques to detect and characterize musical events (typically, the onset, offset and pitch of musical notes). The detection of musical notes is generally based on the harmonic model, that is, a note with a certain pitch f_0 is modeled by a sum of M sinusoidal signals whose frequencies are multiples of f_0 and whose amplitudes and phases are individually defined [3], as in:

$$\sum_{m=1}^{M} a_m \cos(2\pi m f_0 + \phi_m). \quad (1)$$

The output of an automatic transcription algorithm is a set of notes characterized by their pitch (that is, which note in the equal tempered scale is being played), onset (the instant when the note is played) and offset (the instant when the note is damped). Figure 1(a) shows a piano roll representation of a short excerpt. In this representation, boxes represent the detected notes. They are positioned in the row of their corresponding note and they span horizontally from their onset to their onset times. The spectrogram corresponding that same excerpt is shown in Figure 1(b). The goal of our system is to produce the piano roll representation given as input the spectrogram as well as visual information from a camera. This process needs to be carried out in real time which requires all the processing to be causal.



(a) Piano roll.



(b) Spectrogram.

Fig. 1. Piano roll representation and spectrogram for a short excerpt.

When several harmonic signals are summed, the joint detection of their pitches becomes more difficult. This is especially true in cases where the harmonic series of multiple signals overlap, as can be seen in Figure 1(b) where the third harmonic of note F4 coincides with the fourth harmonic of note C4 at 0.55 seconds. This is reflected in the results of most automatic event detection algorithms, which often yield results that correspond to a harmonic or sub-harmonic of the ground truth. As music signals frequently contain combinations of notes with overlapping harmonics this makes the separation and detection of multiple sound sources in music more challenging, in some ways, than speech signals. Another source of errors in event detection algorithms is noise. Most event detection algorithms assume that the analyzed signal contains only audio from musical sources, but that is not true for many cases. There are noises that are inherent to the audio acquisition environment, like the noise of computer fans, heating systems, steps, or even crowds, in the case of live performances. These sources can be identified as musical notes, which may harm the performance of the detection algorithm by increasing the number of false positives.

It is important to note, however, that musical note events are also correlated to certain physical gestures that are performed by the musician. It has been shown that the combination of audio and visual information tends to increase the performance of speech recognition systems [4]. The use of audio and visual information for musical transcription, to the best of our knowledge, was first proposed by Gillet and Richard [5] in a system capable of integrating data acquired from a microphone and a video camera to detect and characterize the hits on a drum kit. In this work we apply a similar approach to the live detection of note events from a vibraphone. Gillet and Richard show that the fusion of audio and video features can lead to better transcription results than the results obtained using solely audio or video. Later, Kapur *et al.* [6] used multiple sensor fusion to detect certain gestures in sitar performances which may easily be confused with each other if only auditory information is considered.

This paper proposes a multi-modal solution for the detection of events in vibraphone performances, aimed at reducing the harmonic errors and the errors due to ambient noise when compared to the traditional audio-only solution. Three different audio signal processing methods were tested: the Non-Negative Least Squares fitting (NNLSQ) [7], the Probabilistic Latent Component Analysis (PLCA) [8], and the auditory-inspired multiple fundamental frequency detection method proposed by Klapuri [9]. The computer vision technique used in the paper relies on a video camera that is mounted on top of the vibraphone to estimate the position of the mallets in relation to the bars. This information is combined with the audio-based results so that positioning the mallet over a bar is a necessary condition for a note to be detected.

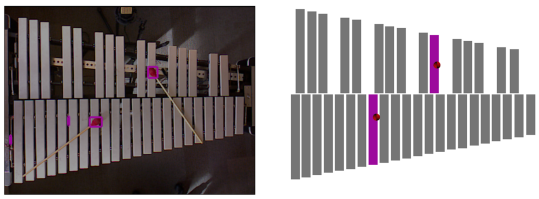
The paper is organized as follows. In Section II, the proposed system and its modules are described. In Section III, the evaluation methods are described and the results are shown. Finally, in Section IV further discussions are conducted and conclusive remarks are presented.

II. System description

The system proposed in this paper relies on both audio and video data, which are jointly analyzed in order to detect when a musician plays a certain note. The audio data is analyzed by a real-time transcription algorithm, which consists of a signal processing front-end and an event detection method. The three signal processing front-end considered were based on NNLSQ [7], PLCA [8], and the method proposed by Klapuri [9]. The event detection method is based on adaptive thresholding. The video data is processed by a computer vision algorithm that is responsible for detecting the position of the mallets over the instruments's body. Since it is necessary to hit a certain bar with the mallet in order to produce sound, the information on the location of the mallet is used to inhibit the detection of notes that are clearly not being hit by the mallet at that moment. This Section is divided in two parts. In the first one, the computer vision algorithm and the hardware requirements for the video camera are described. In the second one, the different audio analysis algorithms are described, as well as how the audio and visual information is integrated.

A. Computer vision

In this work, computer vision techniques are used to track the position of the percussionist's mallet tips relative to the vibraphone bars. This technique allows us to determine which bars of the vibraphone are covered by the mallets and therefore "active". The assumption is that if the mallets are covering a certain bar, then it is likely that the pitch associated with that bar will be heard. Of course there is also the possibility of



a) b)

Fig. 2. Computer Vision Mallet Detection with a) blob detection and b) a virtual scaled copy of the vibraphone.

the mallet covering a bar without the note being played as the performer moves the mallets across the instrument.

Using a VGA webcam (in this work, we used the one that is integrated in the Microsoft Kinect) mounted above the vibraphone and computer vision libraries [10], we are able to track the vibraphone performer's mallet tips based on their color. This process involves filtering out everything except a desired color range, and then performing contour detection on the filtered image. The color filtering process is shown in Figure 3. The color image is first separated into Hue, Saturation, and intensity Value (HSV) components. A binary threshold is applied to each of these components, setting all values within a desired range to 1 and the rest to 0. Recombining these images by performing boolean AND operations yields an image where nearly everything except the tips of the mallet sticks have been removed.

$$x_{binary\ thres} = \begin{cases} 1 & if \quad min < x < max \\ 0 & otherwise \end{cases} \tag{2}$$

Contour detection is performed on the binary image using a technique called border following [11]. This technique involves traversing edges between 0 and 1 in the binary image and creating a sequence. If this sequence returns to it's starting point then a contour has been detected. For our purposes, a bounding rectangle around each contour is returned. Finally, the possible correct results are limited by area, excluding contours that are too large or too small.

Tracking objects with blob detection algorithms returns position data relative to the pixel space of the camera. However, we would like these positions to exist in a virtual space that also includes a 3D model of the vibraphone bars. Using explicit measurements, a scaleable model of the vibraphone was created. Assuming linearity in the position output from the camera, our mallet positions are transformed into our virtual space using the linear normalization shown in Expression 3. Currently, a calibration stage is needed to determine the position of the vibraphone relative to the camera pixel space.

$$p_{norm} = \frac{p_o - p_{min}}{p_{max} - p_{min}} \tag{3}$$

Although this method is simple, sensitive to lighting conditions, and requires manually setting the color of the mallet tips, it has given results that are accurate enough for the purposes of this paper.

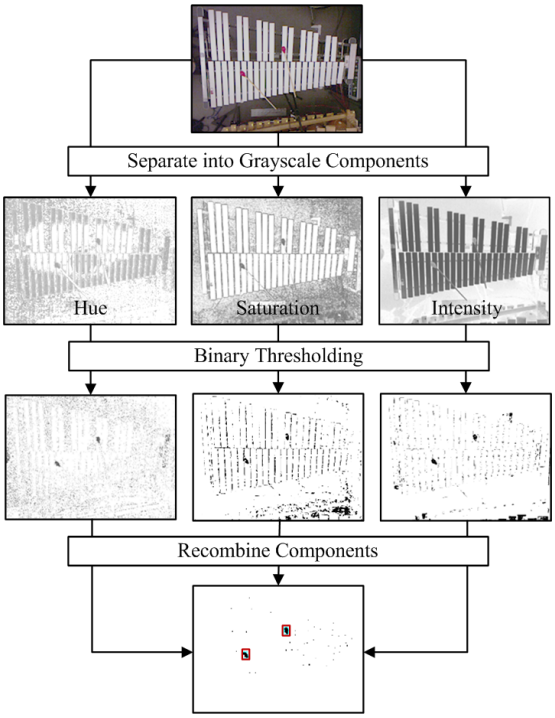


Fig. 3. Computer Vision Flow Diagram

B. Machine listening

The audio analysis techniques we use are based on the assumption that audio signals resulting from the mixing of several different sources are, in terms of physical measures, the sum of the signals corresponding to each individual source. Similarly, the human perception of listening to that sound is essentially the superposition of the sensations triggered when listening to each individual source. Therefore, it is reasonable to assume that, to a certain extent, the phenomenon of detection of sound sources may be described using the following linear model:

$$\boldsymbol{X} = \boldsymbol{B}\boldsymbol{A}. \tag{4}$$

In that model, $\boldsymbol{B}$ is a set of basis functions forming a dictionary, that is, a set of vector representations of the sources to be identified, $\boldsymbol{X}$ is the representation of several measures of the phenomenon in the same domain as $\boldsymbol{B}$, and $\boldsymbol{A}$ is a set of activation coefficients that represent how much each basis function (corresponding to a source) defined in $\boldsymbol{B}$ is active in each measurement. Figure 4 shows an example activation matrix, in which each line represents a different note and each column represents a different time frame. According to the harmonic model described in Expression 1, a particular note tends to have a stationary representation when considering the magnitude of its Discrete Fourier Transform (DFT). This means that the magnitude spectrum is, for harmonic signals, one possible representation that enables the use of the linear model in equation 4.

The assumption of linearity in the harmonic model has been

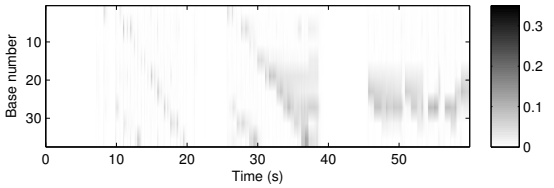


Fig. 4. Example activation matrix.

used successfully in several experiments, both using Non-Negative Matrix Factorization (NMF) [12], [13], in which the basis matrix $\boldsymbol{B}$ is obtained by unsupervised learning, and the NNLSQ approach [14], in which the basis matrix $\boldsymbol{B}$ is obtained previously by means of supervised learning and only the activation matrix is estimated. Although the NNLSQ requires a previous stage of training, it allows causal processing, which is essential for real-time applications. Another way of calculating a conceptually similar activation matrix is by means of the Probabilistic Latent Component Analysis (PLCA) [8]. This technique consists of assuming that both $\boldsymbol{B}$ and $\boldsymbol{A}$ denote probability distribution functions and executing the Expectation-Maximization (EM) technique to obtain them. In the implementation used in this paper, the basis matrix B is obtained in a previous stage from training data and is not adapted (supervised version). Both NNLSQ and PLCA are search algorithms that aim at minimizing the norm of the approximation error $\|\boldsymbol{X} - \boldsymbol{BA}\|$ so their results are similar but are different due to the optimization process utilized.

An alternative method to obtain the activation matrix $\boldsymbol{A}$ is to perform an auditory inspired search for multiple fundamental frequencies, as proposed by Klapuri [9]. This search method incorporates psycho-acoustic knowledge both to design the basis matrix and to obtain the activation of each base. Klapuri's algorithm is based on calculating a salience function for each fundamental frequency candidate, which indicates how much it is active. The most salient candidate is then subtracted from the spectrum and these two steps are iteratively performed until the desired number of notes is obtained. In this paper, the algorithm was adapted so that the activation value for the four notes corresponding to the first four estimated fundamental frequencies are equal to their saliences, and the activation for the other notes are equal to their saliences after the removal of those four notes.

The audio processing algorithms rely on a framewise spectrogram representation, calculated as follows. The input signal is sampled at 48 kHz and processed in discrete frames of 46ms, with a 23 ms overlap. Each frame is multiplied by a Hanning window, zero-padded to twice its original length and is transformed to the frequency domain using the DFT, obtaining X. In order to minimize the influence of of variation in harmonic amplitudes, X is logarithmically scaled as in $Y = \log_{10}(1 + \|X\|)$ [14] (the logarithmic scaling is bypassed when using Klapuri's method, as it already performs spectral normalization [9]). Finally, the frequency domain representation is trimmed in frequency ignoring values ignore values outside the frequency range in which the vibraphone operates.

The basis functions are obtained by taking the spectrogram of a recording containing a single note hit and averaging the first few frames. Only these few frames are used because, in the vibraphone sounds, the harmonics decay quickly. Using more data from a particular note would converge to a spectrogram that would be dominated by the fundamental frequency of the series instead of all the harmonics. The spectrogram is then normalized in order to have unity variance (but not zero mean). The normalization is used in order to ensure that the values in each row of the activation matrix $\boldsymbol{A}$ represent the energy of the activation of the corresponding note.

For the factorization-based methods (NNLSQ and PLCA), a set of additional basis functions, called noise basis functions, are also included in the basis matrix (or dictionary). Noise basis functions are calculated as triangles in the frequency domain, which overlap by 50% and have center frequencies that start at 20 Hz and increase by one octave from one noise base to the next one. This shape aims to give the system flexibility in modeling background noise, specifically non-harmonic sounds, as filtered white noise. This way, background noise is less likely to be modeled as a sum of basis vectors corresponding to notes, that is, background noise is less expected to affect the contents of the activation matrix.

For each incoming frame of audio, the activation vector is calculated. and is then analyzed by an event detection method that yields decisions as to when notes are played. The method works as follows. In order to trigger the detection of a note, the corresponding value in the activation matrix must be over a fixed threshold α and its derivative (that is, $a_{n,t} - a_{n,t-1}$), must be over another threshold β. When a note is found, an adaptive threshold τ value is set to that level multiplied by an overshoot factor γ. The adaptive threshold decays linearly at a known rate θ following the rule $\tau_{t+1} = \tau_t - (1 - \theta)$. A new note may only be detected if its activation value is greater than τ, in addition to the thresholding rules regarding α and β. Figure 5 shows artificial activation values and values for the detection threshold $\max(\alpha, \tau)$. Three positions are marked, showing when false positives are avoided due to the fixed threshold related to α, the adaptive threshold defined by γ and θ and, finally, due to the minimum derivative rule related to β. Additionally to the thresholding related to each individual

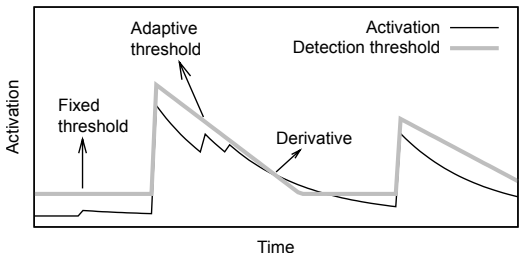


Fig. 5. Example detection by thresholding. False positives are avoided due to the fixed threshold, the adaptive threshold and the derivative rule.

note, the system deals with polyphony by assuming that a

certain activation level only denotes an onset if it is greater than a ratio ϕ of the sum of all activation values for that frame.

If the audio signal passes all tests described above, the estimated bar for the mallet using the computer vision algorithm is used as a necessary condition for detecting a particular note. With this additional condition, it is expected that octave errors and other common mistakes regarding ghost notes can be avoided. It was observed, however, that the signal containing position data was noisy, presenting several false negatives. To deal with that problem, the signal was transformed in a multi-channel binary sinal, where a 1 in channel b means that the mallet is detected over bar b during a certain time frame. The length and hop size of the frames, for this processing, are identical to the ones used in the audio processing algorithm. After this conversion, the binary signal is filtered with a decay filter, defined as:

$$y[n] = \begin{cases} x[n] & x[n] \geq x[n-1] \\ dx[n-1] & x[n] < x[n-1] \end{cases}, \quad (5)$$

where the coefficient d is the decay amount. The filtered signal is thresholded at 0.5 (that is, is equal to 0 if $y[n] < 0.5$ and equal to 1 otherwise). This smoothing process forces the system consider that the mallet is over a bar if it was actually detected there in the last few frames, inhibiting false negatives in that detection. In the next section, the objective evaluation of the effects of using this multi-modal approach for transcription are considered.

III. Evaluation

The experiments described in this section were conducted using audio and video recordings of an acoustic vibraphone. The acoustic data was gathered using a microphone placed in the middle of the vibraphone. The video data was acquired by a videocamera placed on top of the instrument. The recordings intentionally contain harmonically-related notes and heavy use of the sustain pedal. The Signal-to-Noise Ratio varied between -40 dB and 40 dB, both due to the room (heating system, computer fans, etc.), the instrument itself, and artificially-mixed crowd noise. The values for the parameter used in the experiments described here were $\alpha = 0.05$, $\beta = 0.1$, $\gamma = 1.2$, $\theta = 0.1$, $\phi = 0.2$ and $d = 0.9$. These parameters were obtained empirically, by observing the typical values found in the activation matrix and optimizing for better results. The melodies in the evaluation dataset were manually annotated, providing a ground-truth for evaluation. The evaluation data consisted of a total of around 5 minutes of recordings, with 150 annotated notes. The melodic patterns intentionally include polyphony with harmonically-related intervals such as major thirds, fifths and octaves. The number of simultaneous notes varied from one, in simple monophonic phrases, to eight, when the sustain pedal was used. The dataset, as well as the source code used are available upon request.

The transcription algorithm was executed over the evaluation data, yielding a series of symbols. In order to evaluate the performance of the system, the symbols in the automatic transcription are matched to the symbols in the ground truth using a proximity criterion [15]. This matching procedure finds out what note in the automatic transcription was the best attempt (considering a distance measure that accounts for time and pitch deviation) to describe each note in the ground truth. A particular note in the automatic transcription is considered correct if its pitch is equal to the pitch in the ground truth and its onset does not deviate from the ground truth by more than 100 ms. The time tolerance accounts for an inevitable deviation due the framewise nature of the transcription algorithm as well as an inherent human error when the ground truth reference is built. This allows calculating the Recall (R, number of correct events divided by the total number of events in the ground truth), Precision (P, number of correct events divided by the total number of yielded events) and, finally, their harmonic mean called the F-Measure ($F = 2RP/(R+P)$).

This process was executed utilizing the three different audio analysis algorithms (Klapuri's, PLCA and NNLSQ). All three of them were also tested in a configuration that does not use computer vision (blind). We also consider a modification of the algorithm using only vision data (that is, a deaf algorithm), which works by considering an onset as soon as a mallet is placed over a bar and an offset when it is removed from the bar. Two different pieces were used, one of them with heavy use of the sustain pedal and another without using the sustain pedal. The F-Measure for both of them was calculated in each test, and the average of them is reported in Figure 6.

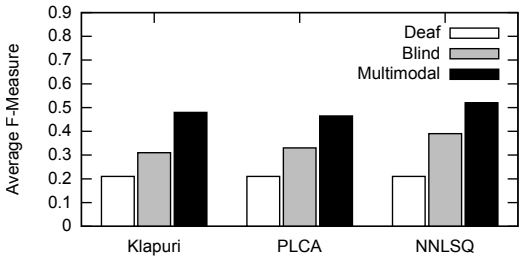


Fig. 6. Average F-Measure for each modification of the proposed algorithm.

The Recall, Precision and F-Measure for each variation of the algorithm are averaged over the two teste tracks and reported in Table I.

TABLE I
RESULTS FOR DIFFERENT ALGORITMS AND PIECES.

Variant	Audio	R	P	F
Deaf	-	0.34	0.15	0.22
Blind	NNLSQ	0.85	0.31	0.39
	PLCA	0.50	0.18	0.26
	Klapuri	0.45	0.25	0.31
Multimodal	NNLSQ	0.50	0.53	0.49
	PLCA	0.35	0.69	0.48
	Klapuri	0.48	0.66	0.48

The results above show that the use of computer vision information has consistently improved the transcription accuracy of the tested systems. Also, regardless of the method used to

obtain the activation matrix $\boldsymbol{A}$, the improvement obtained for the results were roughly the same.

In order to evaluate the changes in the pitch errors due to the use of the vision algorithm, a more detailed evaluation was held. In this evaluation, each note yielded by the transcription system was related to the note that is the closest to it in the ground truth, according to the same distance metric used to obtain the results above [15]. The pitch errors were classified as *octave* errors (if the same pitch class was found, but in a different octave), *tone* errors (if the pitch difference is equal to one or two semitones) and *other* errors (for other pitch deviations). The fraction of errors in each of these categories, considering the blind and multimodal variants of the method that performed best in our experimentos – NNLSQ – as well as the deaf approach, are shown in Figure 7.

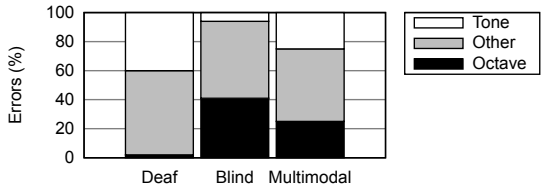


Fig. 7. Pitch errors for deaf and NNLSQ-based variants of the algorithm.

As it can be seen, when the multimodal approach is used, the fraction of *octave* errors was reduced. At the same time, it can be seen that the fraction *tone* errors increases, representing a performance tradeoff indicating further possibilities for improvement.

IV. Conclusion

This paper has presented a method for the multi-modal real time transcription of the vibraphone. The method relies on a causal transcription algorithm that operates in parallel with a computer vision algorithm. The transcription algorithm is responsible to detect, in the audio signal, characteristics that indicate note onsets. The computer vision algorithm yields information on the position of the mallets. Both are combined so that an onset is only detected if the transcription algorithm indicates so and, at the same time, the mallet is over the corresponding bar of the instrument.

Three different signal processing techniques were tested for the audio analysis part of the algorithm: factorization using NNLSQ [7] and PLCA [8], and the multiple fundamental frequency method proposed by Klapuri [9]. The results obtained show that using any of these algorithms result in roughly the same transcription accuracy. Results also show that, regardless of the signal processing used for audio analysis, the use of information from computer vision has consistently improved the transcription accuracy.

The proposed system does not present enough accuracy for applications such as performance analysis or documentation. However, it may still be used in applications that support more errors in the transcription, like chord recognition, automatic harmonization or creative soundscape design.

The are several interesting directions for future work. The computer vision algorithm could be modified tracking algorithm would work in different lighting conditions – an important factor for live performance applications – and in three dimensions, so that the proximity between the mallet and a certain bar can be measured more accurately. We also plan to explore how our systems can be used in applications such as automaic score following, chord detection and harmonization that can take advantage of imperfect transcription results.

Acknowledgment

Tiago Fernandes Tavares thanks CAPES - Coordenacao de Aperfeicoamento de Pessoal de Nivel Superior - for funding. The support of the National Science and Research Council of Canada is gratefully acknowledged. The authors thank Shawn Trail for the assistance with the vibraphone and the research as well as Steven Ness for help with the figures.

References

[1] T.Machover and J.Chung, "Hyperinstruments: Musically intelligent and interactive performance and creativity systems hyperinstruments," in *In Proc. Intl. Computer Music Conference*, 1989.

[2] C. Traube, P. Depalle, and M. Wanderley, "Indirect acquisition of instrumental gesture based on signal, physical, and perceptual information," in *In Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03*, 2003, pp. 42–47.

[3] H. F. Olson, *Music, Physics and Engineering*, 2nd ed. Dover Publications Inc., 1967.

[4] G. Galatas, G. Potamianos, A. Papangelis, and F. Makedon, "Audio visual speech recognition in noisy visual environments," in *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '11. New York, NY, USA: ACM, 2011, pp. 19:1–19:4. [Online]. Available: http://doi.acm.org/10.1145/2141622.2141646

[5] O. Gillet and G. Richard, "Automatic transcription of drum sequences using audiovisual features," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, vol. 3, march 2005, pp. iii/205 – iii/208 Vol. 3.

[6] A. Kapur, G. Percival, M. Lagrange, and G. Tzanetakis, "Pedagogical transcription for multimodal sitar performance," in *International Society for Musical Information Retrieval*, september 2007.

[7] R. J. Hanson and C. L. Lawson, *Solving least squares problems*. Philadelphia, 1995.

[8] P. Smaragdis, B. Raj, and M. Shashanka, "Sparse and shift-invariant feature extraction from non-negative data," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 31 2008-april 4 2008, pp. 2069 –2072.

[9] A. Klapuri, "Multiple fundamental frequency estimation by summing harmonic amplitudes," in *Proc. 7th International Conference on Music Information Retrieval*, Victoria, BC, Canada, Oct. 2006, pp. 1–2.

[10] Intel Corporation, *Open Source Computer Vision Library*. USA: http://developer.intel.com, 1999.

[11] S. Suzuki and K. be, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32 – 46, 1985. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0734189X85900167

[12] N. Bertin, R. Badeau, and G. Richard, "Blind signal decompositions for automatic transcription of polyphonic music: Nmf and k-svd on the benchmark," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 1, april 2007, pp. I–65 – I–68.

[13] S. Phon-Amnuaisuk, "Transcribing bach chorales using non-negative matrix factorisation," in *Audio Language and Image Processing (ICALIP), 2010 International Conference on*, nov. 2010, pp. 688 –693.

[14] B. Niedermayer, "Non-negative matrix division for the automatic transcription of polyphonic music," in *Proceedings of the ISMIR*, 2008.

[15] T. F. Tavares, J. G. A. Barbedo, and A. Lopes, "Towards the evaluation of automatic transcription of music," in *Anais do VI Congresso de Engeharia de udio*, may 2008, pp. 47–51.